

We Claim:

1. In a computer system, a method for transferring data from a data source that produces the data to multiple data sink objects that process the data in the computer system, the method comprising:

receiving the data from the data source;

encapsulating the data into a data object in a memory; and

providing to the multiple data sink objects identification information identifying the data object,

wherein the multiple data sink objects access the data object using the identification information and share the data object with other data sink objects to prevent extraneous copies of the data.

2. The method of claim 1 wherein the step of receiving comprises the step of receiving the data from image acquisition devices.

3. The method of claim 1 wherein the data received from data source includes financial data.

4. The method of claim 1 wherein the data received from data source includes multimedia data.

5. The method of claim 1 wherein the computer system provides a technical computing environment.

6. The method of claim 1 wherein the data sink object performs a function of previewing the data encapsulated in the data object.

7. The method of claim 1 wherein the data sink object performs a function of logging the data encapsulated in the data object to the memory.

8. The method of claim 1 wherein the data sink object performs a function of logging the data encapsulated in the data object to a textual file.

9. The method of claim 1 wherein the data sink object performs a function of logging the data encapsulated in the data object to a binary file.

10. The method of claim 1 wherein the data sink object performs a function of logging a single data sample/frame to the memory.

11. The method of claim 1 wherein the step of providing identification information includes the step of providing to the multiple data sink objects a pointer indicating the location of the data object in the memory.

12. The method of claim 1 further providing a counter for counting a number of the multiple data sink objects that are provided with the identification information of the data object.

13. The method of claim 12 wherein the counter is automatically decreased by the multiple data sink objects when the multiple data sink objects no longer need to access the data object.

14. The method of claim 13 wherein if the counter is decreased to zero, the data object is deleted automatically from the memory.

15. The method of claim 13 wherein if the counter is decreased to zero, the data object is stored persistently and separately in the memory.

16. The method of claim 1 wherein the step of encapsulating comprises allocating the memory to the data automatically.

17. The method of claim 1 wherein the step of encapsulating comprises configuring a maximum amount of memory that all data objects uses at a given instance of time.

18. The method of claim 1 wherein the steps of receiving, encapsulating and providing are performed independently of MATLAB.

19. A data transfer system for transferring data from a data source to multiple data sink objects in a computer system, wherein the data source is coupled to the computer, the system comprising:

an interface for communicating with the data source to receive the data from the data source;

a data processor for encapsulating the data into a data object in the memory; and

a data server for providing to the multiple data sink objects identification information identifying the data object,

wherein the multiple data sink objects access the data object using the identification information and share the data object with other data sink objects to prevent extraneous copies of the data.

20. The system of claim 19 wherein the data server includes a list listing the multiple data sink objects that are registered with the data server.

21. The method of claim 19 wherein the computer system provides a technical computing environment.

22. The system of claim 19 wherein the data server provides a pointer indicating the location of the data object in the memory to identify the data object.

23. The system of claim 19 further comprising one or more data listener object that is registered to the multiple data sink object.

24. The system of claim 23 wherein the data sink object deletes all data listener objects registered with the data sink object when the data sink object is deleted.

25. The system of claim 23 wherein the data sink object notify the data listener when state of the data sink object changes.

26. The system of claim 23 wherein the data sink object notify the data listener when the data sink object is updated with a new data object.

27. The system of claim 19 wherein the data source provides data sequence continuously for a period of time.

28. The system of claim 19 wherein the data source provides a package of data, the package of data being used independently of other packages of data.

29. The system of claim 28 wherein the package of data includes a frame of image data.

30. The system of claim 28 wherein the package of data includes a scan of radar, sensor, or audio data, as well as network data packets.

31. The system of claim 19 wherein the data processor configures a maximum amount of memory that all data objects uses at a given instance of time.

32. The system of claim 19 further comprising a processor for controlling the interface, the data processor, and the data server, wherein the processor is 64 bits or more.

33. The system of claim 19 wherein the interface, the data processor, and the data server are implemented independently of MATLAB.

34. A computer readable medium holding instructions executable in a computer system, wherein the computer system receives data from a data source and transfers the data to data sink objects, comprising:

communicating with the data source to receive the data from the data source;

encapsulating the data into a data object in the memory; and

a data server object providing to the data sink objects identification information

identifying the data object, wherein the data sink objects access the data object using the identification information and share the data object with other data sink objects to prevent extraneous copies of the data.

35. The medium of claim 34 further comprising a data sink listener object that is registered with one or more of the data sink objects.

36. The method of claim 34 wherein the computer system provides a technical computing environment.
37. The medium of claim 35 wherein the data sink listener object performs a task relating to a function of the data sink object.
38. The medium of claim 35 wherein the data sink listener object performs a task relating to a function of the data sink object on a thread of the data server object.
39. The medium of claim 35 wherein the data sink listener object performs a task relating to a function of the data sink object on a thread different from that of the data server object.
40. The medium of claim 34 wherein the data sink object perform a function on a thread of the data server object.
41. The medium of claim 34 wherein the data sink object perform a function on a thread different from that of the data server object.
42. The medium of claim 34 wherein the instructions are run independently of MATLAB.
43. The medium of claim 34 wherein the instructions are originated from code written with C programming language.
44. The medium of claim 34 wherein the instructions are originated from code written with an object-oriented programming language such as C++, C# and Java.